

CLAIM AMENDMENTS

- 1 1. (Currently amended) A method of determining ~~a variable to receive~~ a value
2 change ~~and a value for the~~ a placement variable as part of a local search solution to an
3 integer programming problem that models placement of services of a distributed
4 application onto nodes of a distributed resource infrastructure ~~having polynomial~~
5 ~~terms of at least second order~~ comprising the steps of:
6 selecting an unsatisfied communication constraint that includes a sum of
7 terms, at least some of the terms comprising products of placement variables;
8 creating stores for allowable changes of value for the placement variables in
9 the unsatisfied communication constraint;
10 parsing through the unsatisfied communication constraint by term and for
11 each placement variable in the term updating the stores with a change in the term
12 for each of the allowable changes of the value while maintaining other placement
13 variables constant; and
14 choosing the placement variable to receive the value change and the value for
15 the placement variable based upon the store which meets an improvement
16 criterion.

- 1 2. (Currently amended) The method of claim 1 wherein at least some of the
2 placement variables are ~~comprise~~ a binary variable ~~variables~~.

- 1 3. (Currently amended) The method of claim 2 wherein all of the placement
2 variables are ~~comprise~~ binary variables.

- 1 4. (Currently amended) The method of claim 3 wherein each of the placement
2 variables is associated with no more than a single store.

- 1 5. (Currently amended) The method of claim 3 wherein the step of choosing the
2 placement variable to receive the value change and the value for the placement
3 variable comprises choosing the placement variables to receive the value change

4 based upon which of the stores meets the improvement criterion and flipping the
5 value of the placement variable selected.

1 6. (Currently amended) The method of claim 1 wherein the allowable values for at
2 least one of the placement variables comprise integer values.

1 7. (Original) The method of claim 6 wherein the integer values comprise a subset of
2 possible integer values.

1 8. (Currently amended) The method of claim 7 wherein the subset of the possible
2 integer values comprises the integer values near an existing value for the placement
3 variable.

1 9. (Currently amended) The method of claim 1 wherein the allowable changes in the
2 placement variables do not include tabu values.

1 10. (Original) The method of claim 1 wherein the improvement criterion comprises
2 the store which improves the unsatisfied constraint and which improves an overall
3 solution at least as much as other stores.

1 11. (Original) The method of claim 1 wherein the improvement criterion comprises
2 the store which is not tabu, which improves the unsatisfied constraint, and which
3 improves an overall solution at least as much as other stores which are not tabu.

1 12. (Original) The method of claim 1 further comprising the step of determining that
2 none of the stores improve an overall solution.

1 13. (Original) The method of claim 12 wherein the improvement criterion comprises
2 the store which improves the unsatisfied constraint at least as much as other stores.

1 14. (Original) The method of claim 12 wherein the improvement criterion comprises

2 a random selection of one of the stores which improves the unsatisfied constraint.

1 15. (Currently amended) The method of claim 1 wherein the step of choosing the
2 placement variable to receive the value change and the value for the placement
3 variable is based upon the store which meets the improvement criterion comprises
4 improvement criteria.

1 16. (Currently amended) The method of claim 1 further comprising the steps of:
2 defining a problem model which comprises constraints, wherein the
3 constraints comprise hard constraints and a soft constraint;
4 randomly initializing values for the placement variables, thereby forming
5 assigned values for the placement variables, wherein the assigned values are
6 applied to the constraints to determine ~~the unsatisfied~~ communication constraints;
7 changing the value of the placement variable according to the improvement
8 criterion, thereby modifying the assigned values to form new assigned values; and
9 comparing the constraints to optimality criteria and if the optimality criteria
10 are met, outputting the new assigned values as a near optimum solution.

1 17. (Currently amended) The method of claim 16 wherein, if the optimality criteria
2 are not met, further comprising determining whether to perform an additional
3 iteration and if so, returning to the step of changing the value of the placement
4 variable, wherein the new assigned values become the assigned values.

1 18. (Currently amended) The method of claim 17 wherein, if another iteration is not
2 to be performed, further comprising determining whether to reinitialize the placement
3 variables and if so, returning to the step of randomly initializing the values for the
4 placement variables.

1 19. (Currently amended) The method of claim 18 wherein, if the placement variables
2 are not to be reinitialized, outputting a no solution found message.

20. (Currently amended) A method of determining ~~a binary variable to receive~~ a value change for a binary placement variable as part of a solution to an integer programming problem that models placement of services of a distributed application onto nodes of a distributed resource infrastructure ~~having polynomial terms of at least second order~~ comprising the steps of:

selecting an unsatisfied communication constraint that includes a sum of terms, at least some of the terms comprising products of binary placement variables;

creating a store for each binary placement variable in the unsatisfied communication constraint;

parsing through the unsatisfied communication constraint by term and for each binary placement variable in the term updating the store for the binary placement variable with a change in the term due to flipping a value of the binary placement variable while maintaining other variables constant; and

choosing the binary placement variable to receive the value change based upon the store which meets an improvement criterion.

21. (Currently amended) A method of determining ~~a variable to receive~~ a value change ~~and a value for the~~ a placement variable as part of a local search solution to an integer programming problem that models placement of services of a distributed application onto nodes of a distributed resource infrastructure ~~having polynomial terms of at least second order~~ comprising the steps of:

selecting an unsatisfied communication constraint that includes a sum of terms, at least some of the terms comprising products of placement variables;

creating stores for allowable changes of value for the placement variables in the unsatisfied communication constraint;

parsing through the unsatisfied communication constraint by term and for each placement variable in the term which is encountered for a first time updating the stores with a change in the unsatisfied communication constraint for each of the allowable changes of the value while maintaining other variables constant; and

15 choosing the placement variable to receive the value change and the value for
16 the placement variable based upon the store which meets an improvement
17 criterion.

1 22. (Currently amended) A computer readable memory comprising computer code for
2 implementing a method of determining ~~directing a computer to make a determination~~
3 ~~of a variable to receive a value change and a value for the~~ for a variable as part of a
4 solution to an integer programming problem, ~~having polynomial terms of at least~~
5 ~~second order, the determination of the variable to change and the value~~ the method
6 comprising the steps of:

7 selecting an unsatisfied constraint that includes a sum of terms, at least some
8 of the terms comprising products of variables;

9 creating stores in memory for allowable changes of value for the variables in
10 the unsatisfied constraint;

11 parsing through the unsatisfied constraint by term and for each variable in the
12 term updating the stores with a change in the term for each of the allowable
13 changes of the value while maintaining other variables constant; and

14 choosing the variable to receive the value change and the value for the
15 variable based upon the store which meets an improvement criterion.

1 23. (Currently amended) The computer readable memory of claim 22 wherein at least
2 some of the variables ~~comprise a~~ are binary variables.

1 24. (Currently amended) The computer readable memory of claim 23 wherein all of
2 the variables ~~comprise~~ are binary variables.

1 25. (Original) The computer readable memory of claim 24 wherein each of the
2 variables is associated with no more than a single store.

1 26. (Original) The computer readable memory of claim 25 wherein the step of
2 choosing the variable to receive the value change and the value for the variable

3 comprises choosing the variable to receive the value change based upon which of the
4 stores meets the improvement criterion and flipping the value of the variable selected.

1 27. (Original) The computer readable memory of claim 22 wherein the allowable
2 values for at least one of the variables comprise integer values.

1 28. (Original) The computer readable memory of claim 27 wherein the integer values
2 comprise a subset of possible integer values.

1 29. (Original) The computer readable memory of claim 28 wherein the subset of the
2 possible integer values comprises the integer values near an existing value for the
3 variable.

1 30. (Original) The computer readable memory of claim 22 wherein the allowable
2 changes in the variables do not include tabu values.

1 31. (Original) The computer readable memory of claim 22 wherein the improvement
2 criterion comprises the store which improves the unsatisfied constraint and improves
3 an overall solution at least as much as other stores.

1 32. (Original) The computer readable memory of claim 22 wherein the improvement
2 criterion comprises the store which is not tabu, which improves the unsatisfied
3 constraint, and which improves an overall solution at least as much as other stores
4 which are not tabu.

1 33. (Original) The computer readable memory of claim 22 further comprising the step
2 of determining that none of the stores improve an overall solution.

1 34. (Original) The computer readable memory of claim 33 wherein the improvement
2 criterion comprises the store which improves the unsatisfied constraint at least as
3 much as other stores.

1 35. (Original) The computer readable memory of claim 33 wherein the improvement
2 criterion comprises a random selection of one of the stores which improves the
3 unsatisfied constraint.

1 36. (Currently amended) The computer readable memory of claim 22 wherein the
2 step of choosing the variable to receive the value change and the value for the
3 variable is based upon the store which meets the improvement criterion comprises
4 improvement criteria.

1 37. (Currently amended) The computer readable memory of claim 22 further
2 comprising the steps of:
3 defining a problem model which comprises constraints, wherein the
4 constraints comprise hard constraints and a soft constraint;
5 randomly initializing values for the variables, thereby forming assigned values
6 for the variables, wherein the assigned values are applied to the constraints to
7 determine ~~the~~ unsatisfied constraints;
8 changing the value of the variable according to the improvement criterion,
9 thereby modifying the assigned values to form new assigned values;
10 comparing the constraints to optimality criteria and if the optimality criteria
11 are met, outputting the new assigned values as a near optimum solution.

1 38. (Previously Presented) The computer readable memory of claim 37 wherein, if the
2 optimality criteria are not met, further comprising determining whether to perform an
3 additional iteration and if so, returning to the step of changing the value of the
4 variable, wherein the new assigned values become the assigned values.

1 39. (Previously Presented) The computer readable memory of claim 38 wherein, if
2 another iteration is not to be performed, further comprising determining whether to
3 reinitialize the variables and if so, returning to the step of randomly initializing the
4 values for the variables.

1 40. (Original) The computer readable memory of claim 39 wherein, if the variables
2 are not to be reinitialized, outputting a no solution found message.

1 41. (Currently amended) A computer readable memory comprising computer code for
2 ~~implementing a method of determining directing a computer to make a determination~~
3 ~~of a binary variable to receive a value change for a binary variable~~ as part of a
4 solution to an integer programming problem ~~having polynomial terms of at least~~
5 ~~second order~~, the method determination of the binary variable comprising the steps
6 of:

7 selecting an unsatisfied constraint that includes a sum of terms, at least some
8 of the terms comprising products of binary variables;

9 creating a store in memory for each binary variable in the unsatisfied
10 constraint;

11 parsing through the unsatisfied constraint by term and for each binary variable
12 in the term updating the store for the binary variable with a change in the term due
13 to flipping a value of the binary variable while maintaining other variables
14 constant; and

15 choosing the binary variable to receive the value change based upon the store
16 which meets an improvement criterion.

1 42. (Currently amended) A computer readable memory comprising computer code for
2 ~~implementing a method of determining directing a computer to make a determination~~
3 ~~of a variable to receive a value change and a value for the for a variable~~ as part of a
4 solution to an integer programming problem ~~having polynomial terms of at least~~
5 ~~second order~~, the method determination of the variable to change and the value
6 comprising the steps of:

7 selecting an unsatisfied communication constraint that includes a sum of
8 terms, at least some of the terms comprising products of variables;

9 creating stores in memory for allowable changes of value for the variables in
10 the unsatisfied constraint;

11 parsing through the unsatisfied constraint by term and for each variable in the
12 term which is encountered for a first time updating the stores with a change in the
13 unsatisfied constraint for each of the allowable changes of the value while
14 maintaining other variables constant; and
15 choosing the variable to receive the value change and the value for the
16 variable based upon the store which meets an improvement criterion.